Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Platform and direct exchange-based mechanisms for resource sharing in distributed manufacturing: a comparison

Ádám Szaller[a,b], Botond Kádár[b] (1)

[a]Centre of Excellence in Production Informatics and Control, Institute for Computer Science and Control, Budapest, Hungary
[b]Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest, Hungary

The operation of production facilities is shifting from centralized organizations towards decentralized networks. The paper investigates and compares alternative mechanisms for resource sharing in distributed manufacturing. Specifically, with the same underlying assumptions, a platform and a direct exchange-based model are presented and examined. The models have in common that resource assignment decisions are made ultimately by the autonomous facilities, also based on trust they maintain towards each other. Agent-based simulation is used to compare the two mechanisms with respect to utilization rate, service level and communication load. The findings can be applied in the design of crowdsourced manufacturing platforms.

Distributed manufacturing, Manufacturing network, Resource sharing

1. Introduction

The phases of globalization changed existing paradigms of manufacturing. First large companies, and then, Small and Medium Enterprises shifted from centralized manufacturing organizations towards decentralized networks, with the aim of maintaining their market role [1].

Crowdsourced manufacturing was proposed by the International Electrotechnical Commission: the essence of the concept is to share resources with each other via a *platform* to utilize them in more efficient and robust ways [2]. This concept is mainly applicable for assembled products demanding expensive machines, inspection processes, technologies of production, quality control, etc. For Build-to-Order companies, keeping excess production capacities in order to meet customer deadlines is a must, and sharing resources via crowdsourcing can be a viable solution in reaching high resource utilization levels [3].

Resource sharing in production networks – with or without a central platform – has been investigated by researchers in the past years. In [4], resource sharing in case of galvanizing lines is examined using simulation modeling, and the authors show that an increase in information exchange is not necessarily beneficial for all participating companies. In [5], the stability of request-offer matching is analyzed in crowdsourcing. [6] highlights that for each participant (requesters, offerors, platform), it is important to consider the cost of matching. Beyond communication, significant effort was dedicated to the development of specific decision-making mechanisms as well. In [7], a multi-criteria decision model is proposed, using a dynamic decision approach for supplier and business partner evaluation in a collaborative network. Robustness of capacity allocation was studied within dynamic production networks in [8], with the aim of finding an optimal capacity allocation to the production processes of a given network for the desired robustness.

It is essential that resource sharing organizations should be encouraged to keep their promises and to have commitments to their plans and goals. Trusting in each other's promises is one of the main pillars of collaboration, which also helps to increase the service level of the participants and to reduce risks [9] [10]. Nevertheless, computational models applying trustfulness in decision-making are mainly used only in computer technology and online commerce [11]. Very few examples can be found where these aspects are examined in the manufacturing area, for example [12] presents a Cloud Manufacturing model that considers direct, indirect and third-party experiences. In [13], a multi-criteria variable weights decision-making approach based on trust and reputation in supply chains is proposed. However, the above mentioned two models focus on the detailed trust model and neglect the capacity constraints of the participants.

Based on the examples from the literature, it seems that resource sharing generally improves the participant's resource utilization level. The current paper investigates and compares two different ways on *how* it is more valuable to share resources: *a)* by using a direct exchange-based mechanism or *b)* by joining an intermediate platform responsible for matching requests with the offers. By modelling the same set of facilities loaded with an order stream having the same parameters, a fair comparison is made that is unique in the literature. The performance of the mechanisms is assessed considering three different viewpoints: *a)* average resource utilization, *b)* average service level, and *c)* communication load. The compared approaches consider trust-related aspects as well, which are usually neglected in resource sharing case studies or investigated without taking capacity constraints into account.

In the first reference model – presented in [9], referred to as "Model A" – participants communicate and coordinate their actions directly with each other: they are sending resource requests to all the other organizations who are part of the federation. When receiving a request, all the participants being able to complete the request are applying for the given job and send offers to the requester.

The basis of the second model ("Model B") was introduced in [14], and the theory of a modified and extended version was published in [15]. In this case, the participants are sending their resource offers and requests to a central platform, which does the matching: sends the appropriate offers to the requester facility from the already received ones. Since the platform is aware of all requests and offers, it can combine several offers to complete one request, and is able to optimize e.g., logistic routes on a global level.

In the following sections, the basic concepts are introduced in connection with the two resource sharing methods, and the most important similarities and differences between the two approaches are highlighted. Finally, the results of experiments using agent-based simulation of both models are examined and presented, comparing the performance of the two approaches.

2. Description and comparison of the models

For easier understanding, concepts are clarified in connection with the compared resource sharing methods. Most of them are the same in both models – except in the cases where the difference is highlighted.

In the resource sharing models, a *facility* is an autonomous decision-maker, who can communicate with other facilities, make decisions, and create plans for the future based on its own KPIs, goals, and resources. It can be an offeror and a requester at the same time: the denomination depends on the role in the specific interaction. When an offeror facility signs a contract, then it is expected to: (1) complete the job, (2) finish it on time, and (3) complete it with the expected quality. In both models, facilities are rating their partner's performance based on these three aspects and consider the ratings when selecting from offers.

In Model A, as facilities are communicating directly with each other and thus know the identity of the offerors, they can calculate with *trust* (subjective rating based on direct interactions) and *reputation* (public rating updated by the FC, containing other facilities' ratings, too). In Model B, the identity of the winners becomes known only after the decision was made, in order to prevent facilities from discovering each other's resources with the aim of creating a competitive advantage. Thus, subjective trust values cannot be considered in decision-making, only reputation, which is calculated in the same way as in Model A.

Both trust and reputation are cumulated values calculated based on ratings given by the requester facility about each interaction, as discussed in [15]. The basis of the rating is the percentage of lateness in completing the job. This is changed according to quality aspects or replaced if the job was cancelled by the offeror. A modified exponential smoothing function is applied in both models to give older feedbacks smaller weights than more recent ratings.

Collaborating facilities form a *federation*. Collaboration is only possible between federation members but entering or exiting the federation is allowed anytime. The central unit of the federation is the *Federation Centre (FC)*. For calculating and updating reputation values and ensuring their public availability a central platform is necessary in both cases; nevertheless, its main role is different in the two models. In Model A, it manages the entries and exits from the federation. In Model B, besides these activities, the facilities send all the resource offers and requests to the FC, which dynamically matches them and manages the contracting, as well.

Facilities receive customer *orders* from outside the federation. One order represents one *job*, which is determined by its resource requirements: type (e.g., CNC machine), quantity (e.g., 3 pieces), earliest start time and due date. The following assumptions are taken: to fulfil a job, the *resource load* of the job has to be provided, which is calculated by multiplying the resource quantity with the difference between the due date and the earliest start time. This means, with more available resources, the job could be completed in less time. When facilities receive a customer order, they might not have the appropriate or sufficient resources to complete it. In this case, they send *requests* to all the other federation members (Model A) or to the FC (Model B). A request contains all the resource requirements mentioned in the case of a job, and in Model B, the maximum number of fragments the request can be divided into. In Model A, when a facility receives a request, it checks the already planned works for the future. If the appropriate resources (same type, sufficient quantity in the required interval) are available, it sends an *offer* to the requester, which can choose the best based on its decision mechanism.

In Model B, facilities send offers to the FC regularly – these offers are about outsourcing a certain resource quantity from a given type for a specific time interval (taking an internal safety margin into consideration). Receiving a request, the FC checks whether there is a match between the new request and the active offers. If the FC receives an offer, it checks whether some of the non-fulfilled requests can be completed with the new offer (or by combining the new offer with the earlier ones). If the FC finds a match, it notifies the requester about *all the possible offer combinations* fulfilling the request. In other words, the FC provides access to a capacity sharing community – it operates like a dynamic augmentation of the facility, which can be extremely useful in case of fluctuating customer orders. It is important to mention that the goal for the FC is not to select the best solution, but to pre-filter the offers that are meeting the resource constraints of a request, by integrating reputation values as well. This way, the FC limits the communication and decision space, and expands it at the same time, since it can find suitable offer combinations. Facilities could also set a minimum reputation value below which they do not want to receive offers. The FC considers all the offers sent by facilities above this reputation level in the matching process but leaves the decision to the requester facility, who takes the prices into consideration, as well.

The differences between the two approaches are summarized in Table 1.

Table 1. Differences between the compared approaches.

Comparison	Model A	Model B
Role of FC	Managing entries and exits, updating reputation values	Same as in Model A + matching, managing contracts
Communication, information sharing	Directly with each other	Only with the FC
Anonymity	Facilities know the offer parameters and the identity of the offerors	Facilities know the offer parameters without sender identity, and the identity of the winner
Dividing requests into fragments	Requesters outsource all the fragments separately	FC generates all possible offer combinations
Computational load	Requesters receive only the suitable offers	The FC checks all the received offers, requests
Decision-making	Reputation and trust are considered	Reputation is considered, trust is not
Optimizing on a global level	Not possible	Possible

In Model A, the facilities are exchanging resource information about themselves directly with each other in case of each request or offer. In Model B, information is shared only with the FC, and, in addition, the requester facility knows only the identity of the winner. In the first case, facilities must trust everyone else in the federation; in the second case, it is enough to trust the FC.

In Model A, a facility tries to divide the offer into a feasible number of equal-sized fragments in case of receiving no offers for the whole request, as it does not know the free capacities of the others and cannot adjust the request sizes accordingly. In Model B, the requester can set the maximum number of offers that can be combined to fulfil the request, and the FC sends all the possible offer combinations accordingly.

Other aspects are the computational and communication load of matching. In Model A, requests are sent to all the facilities in the federation, and only the facilities that have the proper resources to fulfil the request, will send an offer. In case of fragmented requests, each fragment is treated separately: facilities must check their available resources for each of them and send separate offers to the requester. In contrast, in Model B, requests and offers are sent only to the FC, which performs the matching. Requests do not have to be divided into pre-defined fragments, because the matching function can combine the received offers based on the maximum number of fragments. Faster reaction time can be achieved since a facility does not have to wait for several answers, it is enough to get feedback from the FC. However, due to offer anonymity, facilities cannot calculate with trust values (reputation values are available in both cases).

Naturally, model B is more vulnerable in the case a problem occurs with the FC. In such a case, the whole resource sharing process fails, and the facilities could try to contact each other directly. In Model A, only the reputation values are lost, subjective trust values are still available. Loss of access to the updated member list of the federation affects both cases. A substantial difference between the two approaches is that only Model B provides the possibility to optimize on a global level, as the FC is aware of all requests, offers and contracts created in the federation.

In both models, only discrete resources are considered, and jobs are completed on a First Come First Served basis. Setup times are not explicitly modelled, they are included in the processing times of the jobs. Distance between facilities and their resource types are also considered, but the authors do not deal with the modelling of maintenance periods, shifts, premises, buffers, etc. It is assumed that the participants are honest and do not try to manipulate the participants by sending false messages. Security issues of the communication mechanisms are also not focus points here. Cancelling contracts can occur in both models – in this case the facility receives a reputation penalty.

3. Experiments

To compare the two models, experiments with agent-based simulation were performed in AnyLogic, where facilities and the FC were modeled with agents. In the experiments, all facilities receive an order stream generated within the model and perform different functions triggered by specific events: for example, the offer-making function is called when some resources are released. The FC also performs pre-defined functions when receiving a message (request, offer or acceptance/cancellation) and stores not matched requests and offers in its own database.

Regarding the input, in both models, the same facilities with the same resources were loaded with the same average order size and arrival rate. The most important parameters of the experiments are presented in Table 2. For the parameters determined by using a truncated normal distribution, the mean and sigma values are included in the table (for the constant ones, sigma is 0). In these cases, the difference between the lower and upper bounds of the distribution from its mean is sigma/2. The parameters relevant in only one of the two models are marked with a superscript.

In the experiments, out of 10 facilities, 6 are *non-reliable,* and 4 are *reliable*. The job completion times are also normally distributed: for non-reliable facilities, the mean of the distribution is shifted, creating a lower chance for them to finish the undertaken jobs in time. In the current stage of the research, reliable facilities do not have higher prices than non-reliable ones. Facilities differentiate between their partners based on trust and reputation values: reliable partners having higher ratings. In the experiments, 20 different resource types were initiated: one facility had 10 to 20 types of them with the amount 8 to 12.

As indicated in Table 2, each facility received an order every 1.5 days, required 8 resource units for 20 time units from a specific resource type. To reduce the administrative costs of contracting and the computational load of the FC, matching is only possible between exactly one request and a maximum of three offers in Model B. Similarly, requests can be divided by requester facilities into three parts in Model A. In Model B, facilities sent offers to the FC about their free resources with a lookahead for the next 40 time units. In all cases, the experiments were run for 500 time units: based on observations, the investigated KPIs do not change after this time in an unexpected way. Since some of the parameters are normally distributed, 10 experiments were performed for each parameter set, and the average of the values is presented in the diagrams. The three examined KPIs were *average resource utilization, average service level* of federation members and *communication load* (defined below in detail). In both cases, 48×10 experiments were performed while increasing the number of federation members from 3 to 50 facilities, with the aim of investigating the effect of federation size on the KPIs.

Table 2. Input parameters for experiments.

Parameter	Mean	Sigma	Unit
Initial reputation^{A,B} and trust^{A}	80	0	-
Incoming order arrival rate	1.5	0	1/mtu*
Incoming order length	20	5	mtu
Incoming order resource quantity	8	2.7	-
Max. number of offer fragments	3	0	-
Planning horizon^{B}	40	8	mtu
Simulation time	500	0	mtu

*model time units

3.1. Average resource utilization

To compare the federation performances in the two models, resource utilization of the federation members was sampled in every time unit. The average of these values after each simulation run was calculated by neglecting the first 30 values out of 500 (run-up phase). The confidence intervals on 95% confidence level in case of the remaining 470 values obtained from a specific simulation run were between 2~3% (4~5% in federation sizes under 10). As one can see in Figure 1, in both models, average resource utilization increases until the federation size reaches 10. The difference between the two models is approx. 30% regardless of the federation size. This is mainly due to the effect of the FC, which finds matching offers for requests in a more efficient way and divides requests into fragments adjusted to already received offers. In contrast, in Model A, facilities try to outsource equal-sized fragments which limits the solution space.

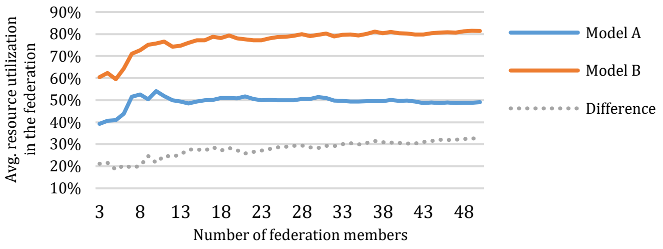

Figure 1. Difference between average resource utilization in the models.

3.2. Average service level

The performance of the resource sharing approaches could be measured by comparing their ability to find more reliable partners to work with. Outsourcing jobs to more reliable partners results in less delay in job completion times: average service level is used to highlight this setting. This KPI is calculated by measuring percentage lateness in case of all completed jobs, subtracting it from 100%, and recording the average of these values after each simulation run. The results show that Model A performs better: after reaching federation size 10, the difference between the two approaches is approx. 4~5% (see Figure 2). The confidence intervals on 95% confidence level are between 0.1%~1%. Since the average resource utilization is much lower in Model A, the reliable partners are not as much loaded as in Model B. Thereby, the facilities have a better chance to choose a more reliable partner in the first case. This means a kind of trade-off between resource utilization and service level: if the goal is to maximize resource utilization, offers from non-reliable partners have to be accepted as well; but if the aim is to reach a high service level, a limit could be defined for trust/reputation values of possible partners. This setting is also included in the models, but its effect is planned to be investigated in further research. The cause of the non-monotonous trend in the results is that increasing the number of federation members was done by creating facilities with the same parameters as the first 10 in the same order. For example, in case of 16 federation members, 12 were non-reliable and only 4 reliable.

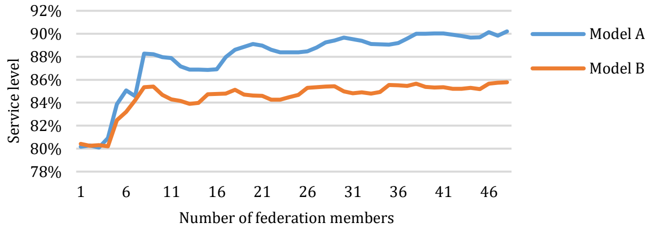

Figure 2. Difference between service levels in the two models.

3.3. Communication load

In the third experiment, the number of messages was investigated. As described in Section 2, the two models are using different communication mechanisms. Message types in Model A are the following: *original/divided requests* (sent to all federation members), *offers* (sent in response to a request), *notifying winner(s) and loser(s)* in response to offers, *contracts* between the requester and offeror directly, and *contract cancelling* messages. *Ratings* about a contract are also sent to the FC.

In case of Model B, only *original requests* and *offers* are sent to the FC. Only the winners are notified after matching – the offers (and requests) that are not matched are deleted automatically from the offer database when they expired. *Notifying winner(s)* and *contracting* are made through the FC *Cancelling* and *rating contracts* are made through the FC, also.

Figure 3 highlights the communication load which increases heavily in Model A, even more than linearly, when the federation grows. A second order polynomial trendline was fit to the results in case of Model A, and a linear trendline in case of Model B. For each experiment, the difference between the two models depends on the parameters of the incoming orders (arrival rate, resource quantity, length) and the parameters of the facilities, also.

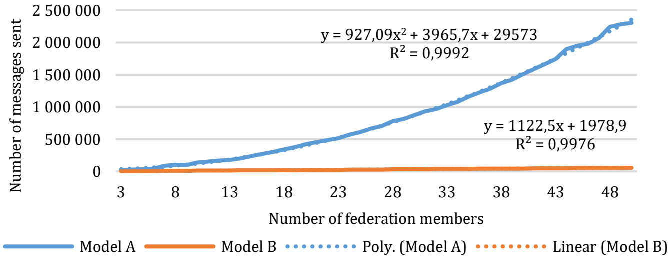

Figure 3. Difference between communication load in the two models.

Conclusions and future work

In the paper, two manufacturing resource sharing approaches were compared: Model A, where resource sharing is made by a direct exchange-based mechanism, and Model B, where resource requests and offers are matched by an intermediate platform. Agent-based simulation experiments have shown that the difference between average resource utilization is approx. 30% in favour of Model B, due to the more complex matching logic of the FC. However, the average service level is approx. 4~5% higher in Model A, since lower resource utilization causes that reliable facilities have more free capacities that can be requested by others. Communication load is one order of magnitude lower in Model B, because the facilities are sending messages only to the FC instead of each other.

The novelty of the research presented in this paper is the unique comparison of two different resource sharing approaches, which both consider trust aspects in decision-making while taking capacity constraints into account, as well. It is important to mention that while a platform-based solution could improve federation performance from different aspects, it does not take away the possibility of autonomous decision-making from the participants. In addition, a platform can limit the decision space by selecting the appropriate offers and widen it at the same time in useful directions (for example with combining offers).

In future work, the model will be extended by more complex orders containing interdependent jobs that may require different types of resources. A multi-criteria decision-making algorithm will also be developed to model choosing from resource offers in a more complex way, and reliable facilities having premium prices will also be considered.

Acknowledgements

This work has been supported by the National Research, Development and Innovation Office, grant no. SNN 129178 and by the European Commission through the H2020 project EPIC (https://www.centre-epic.eu/) under grant No. 739592.

References

[1] Lanza, G., Ferdows, K., Kara, S., Mourtzis, D., Schuh, G., Váncza, J., Wang, L., Wiendahl H.-P., 2019, Global Production Networks: Design and Operation, CIRP Annals, 68 (2): 823–841.

[2] International Electrotechnical Commission (IEC), 2015, Factory of the Future: IEC Whitepaper, 20-23.

[3] Kaihara, T., Katsumura, Y., Suginishi, Y., Kádár, B., 2017, Simulation Model Study for Manufacturing Effectiveness Evaluation in Crowdsourced Manufacturing, CIRP Annals, 66 (1): 445–448.

[4] Freitag, M., Becker, T., Duffie, N. A., 2015, Dynamics of Resource Sharing in Production Networks, CIRP Annals, 64 (1): 435-438.

[5] Chida, T., Kaihara, T., Fujii, N., Kokuryo, D., 2019, Stability Analysis on Resource Matching in Crowdsourced Manufacturing, Procedia CIRP, 81: 405-410.

[6] Chen, Q., Xu, Q., Wu, C., 2019, Dynamic Matching in Cloud Manufacturing Considering Matching Costs, Complexity, vol. 2019

[7] Arrais-Castro, A., Varela, M. L. R., Putnik, G. D., Ribeiro, R. A., Machado, J., Ferreira, L., 2018, Collaborative Framework for Virtual Organisation Synthesis Based on a Dynamic Multi-criteria Decision Model, International Journal of Computer Integrated Manufacturing 31 (9): 857–868.

[8] Scholz-Reiter, B., Wirth, F., Makuschewitz, T., Schönlein, M., 2011, Robust Capacity Allocation in Dynamic Production Networks, CIRP Annals, 60 (1): 445–448.

[9] Szaller, Á., Egri, P., Kádár, B., Trust-based Resource Sharing in Distributed Manufacturing, International Journal of Computer Integrated Manufacturing, 2020, 33 (1): 1-21.

[10] Váncza, J., Márkus, A., 2000, An Agent Model for Incentive-based Production Scheduling, Computers in Industry, 43: 173–187.

[11] Pinyol, I., Sabater, J., 2013, Computational Trust and Reputation Models for Open Multi-agent Systems: A Review, Artificial Intelligence Review, 40: 1–25.

[12] Yan, K., Cheng, Y., Tao, F., 2016, A Trust Evaluation Model Towards Cloud Manufacturing, The International Journal of Advanced Manufacturing Technology, 84: 133–146.

[13] Chang, L., Ouzrout, Y., Nongaillard, A., Bouras, A. Jiliu, Z., 2014, Multi-criteria Decision Making Based on Trust and Reputation in Supply Chain, International Journal of Production Economics, 147 (Part B): 362–372.

[14] Kádár, B., Egri, P., Pedone, G., Chida, T., 2018, Smart, Simulation-based Resource Sharing in Federated Production Networks, CIRP Annals, 67 (1): 503–506.

[15] Szaller, Á., Pedone, G., Egri, P., Szalóki, Á., Nick, G., 2020, A Mutualistic Framework for Sustainable Capacity Sharing in Manufacturing, Procedia CIRP, 93: 938-943.